UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/671,012	09/25/2003	Kavitha Srinivas	YOR920030251US1 (16768)	7874

23389 7590 05/21/2008

SCULLY SCOTT MURPHY & PRESSER, PC
400 GARDEN CITY PLAZA
SUITE 300
GARDEN CITY, NY 11530

EXAMINER	
DAO, THUY CHAN	

ART UNIT	PAPER NUMBER
2192	

MAIL DATE	DELIVERY MODE
05/21/2008	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No.	Applicant(s)	
	10/671,012	SRINIVAS ET AL.	
	Examiner	Art Unit	
	Thuy Dao	2192	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE _03_ MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *25 February 2008*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1,4-7,11,12,15,17-19 and 21* is/are pending in the application.

4a) Of the above claim(s) *2,3,8,10,16,20 and 22* is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1,4-7,11,12,15,17-19 and 21* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *25 September 2003* is/are: a)☒ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☐ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

1. This action is responsive to the amendment filed on February 25, 2008.

2. Claims 1, 4-7, 11-12, 15, 17-19, and 21 have been examined.

Response to Amendments

3. In the instant amendments, claims 6 and 7 have been amended.

4. The objection to claim 6 is withdrawn in view of Applicants' amendments.

Claim Objections

5. Claims 1, 4-7, 15, and 19 are objected to because of minor informalities.

Claims 1 and 15, line 2, the phrase is considered to read as - -...for monitoring [[the]] behavior of a running computer program ...- - ;

Claims 4-7 are considered to read as --A software tool according to Claim 1--; and

Claim 19 is considered to read as - -A program storage device according to Claim 15 [[16]], wherein said method steps ...- -.

Appropriate correction is required.

Response to Arguments

6. Applicants' arguments have been fully considered. However, they are not persuasive.

7. Rejections under 35 USC 102(e) (Remarks, pp. 8-16):

As an initial matter, the examiner notes that Applicants' argument amounts to mere statements and assertions, which are not related to the text/code portions extracted from the references used to reject the claimed limitations as set forth in the Non-Final Office action mailed November 23, 2007.

MPEP, Appendix R - Patent Rules, Action by Applicant and Further Consideration, section 1.111 (b), lines 2-5 expressly states:

> "The reply by the applicant or patent owner must be reduced to a writing which distinctly and specifically points out the

> supposed errors in the examiner's action and must reply to every ground of objection and rejection in the prior Office action" (emphasis added).

As acknowledged by the Applicants (Remarks, page 9), the applied figure/text portions in the previous Office action have been related to at least Figures 12, 14-17, and associated text.

However, Applicants' arguments directed to Figures 2-4 (Remarks, pp. 10-11), general assertions (page 12), briefly mentioned to Figure 12 and 14-17 (page 12, only the last paragraph), general assertions (pp. 13-15), and briefly mentioned to Figure 14 (page 15, only the last paragraph), which did not distinctly and specifically point out the supposed errors in the prior Office action.

After further consideration, the examiner would like to response and address all limitations of claim 1 (and recited in such manners in claim 15) for providing antecedent basis for all grounds of rejection.

Morshed explicitly teaches a program storage device readable by machine and *a software tool containing machine readable instructions stored on a physical medium for monitoring behavior of a running computer program* (e.g., FIG. 12, col.21: 6-24, a VM runtime system runs and monitors a program comprising Java byte code,

also FIG. 12, Monitoring DLL 414 monitoring said running program and sending message stream to Analyzers/Viewers 416 and/or Message Data Storage 417, col.21: 55-67)

for code patterns (e.g., FIG. 14, col.23: 1-64, Java classes as code patterns "selected from a group comprising best practice patterns" as recited in claim 1, lines 16-17;

col.13: 40-56: "scope change" code blocks as code patterns "selected from a group comprising ... problematic coding patterns" as also recited in claim 1, lines 16-17,

wherein said "scope change" code blocks (problematic coding patterns) may create memory leaks, col.19: 13-33)

that violate a given set of coding rules (e.g., col.13: 43-49, coding rule as must free allocated memory to avoid memory leak;

col.17: 25-28: coding rules have been violated and run time errors occur;

col.20: 60-62: violating coding rules such as "improper parameter lists passed to functions, methods, or procedures, use of uninitialized variables";

FIG. 64, col.75: 63 – col.76: 23: results of violating coding rules displayed in a bug report)*, the software tool comprising:*

a pattern detector manager (e.g., FIG. 4, Code Instrumentation 50, col.8: 4-28; col.6: 63 – col.7: 2) *including machine readable instructions for inserting into a running computer program a plurality of entry breakpoints* (e.g., FIG. 14, blocks 442, 448, 452, 456, 460, 464, col.23: 1 – col.24: 11)*,*

automatically, with little or not intervention from a user (e.g., col.20: 14-19; col.20: 63 – col.21: 5)*,*

each of said entry breakpoints being associated with one of a plurality of defined coding patterns (e.g., FIG. 14, common coding patterns of Java classes such as Method Entry (block 442), End of Method (block 446), Throw instruction (block 454), Method Call (block 462), Abort (block 448), col.23: 1-64)*; and*

a plurality of pattern detectors, each of the pattern detectors being associated with one of said defined coding patterns (e.g., FIG. 12, Instrumentation DLL 410, Monitoring DLL 414, col.21: 43-67, wherein said components insert instrumentation code associated with defined best practice coding patterns (as Java classes, col.23: 1-64) and/or defined problematic coding patterns (as "scope change" code blocks, col.13: 40-56; col.19: 13-33)*,*

including machine readable instructions, and being invoked by the pattern detector manager, after one of the entry breakpoints associated with the coding pattern associated with said each of the pattern detectors, is reached in the computer program (e.g., FIG. 14, block 442 Instrument Method Entry, col.23: 6-13;

FIG. 15, details of the instrumentation for a method entry, col.24: 11-62)

for determining whether the computer program violates the coding pattern associated with said each of the pattern detectors (e.g., FIG. 14, block 448, Instrument for Abort to determine any violations in Java classes (best practice patterns), col.23: 14-22;

col.19: 13-33, instrumenting "scope change" code blocks (problematic coding patterns) to detect memory leaks, and col.13: 40-56)

by inserting into the program at least one further breakpoint for identifying a respective step in the program that is part of the coding pattern associated with said one of the entry breakpoints (e.g., FIG. 14, block 448, Instrument for Abort as at least one further breakpoint associated with said method entry breakpoint in block 442, col.23: 14-22;

FIG. 16, details of the instrumentation for an abort step, col.24: 63 – col.25: 37;

col.66: 20-31, in the debugging mode "step", instrumenting entry breakpoint and subsequent breakpoints, col.66; 20-31;

FIG. 14, after Instrument Method Entry at block 442, inserting further breakpoints at blocks 452, 456, 460, 464, col.23: 36 – col.24: 11)*;*

wherein the plurality of defined coding patterns is selected from a group comprising best practice patterns (e.g., FIG. 14, Java classes as best practice patterns may be instrumented and monitored, such best practice patterns including common

coding patterns such as method entry (col.23: 6-13), throw instruction (col.23: 44-51), exit point of said method (col.23: 52-64), ...) *and*

[selected from a group comprising] problematic coding patterns (e.g., col.13: 40-56, "scope change" code blocks as problematic coding patterns may be also instrumented and monitored to detect memory leaks, and col.19: 13-33).

8. Rejections under 35 USC 103(a) (Remarks, pp. 16-24):

a) Morshed and Bates (pp. 16-20):

As set forth in (7) above, Morshed explicitly teaches the claimed limitations.

In claim 21, Morshed does not explicitly disclose *monitors the computer program for the non-occurrence of any one of a second set of defined conditions, the non-occurrence of which violates the coding rule associated with said each of the pattern detectors*.

However, in an analogous art, Bates further discloses *monitors the computer program for the non-occurrence of any one of a second set of defined conditions, the non-occurrence of which violates the coding rule associated with said each of the pattern detectors* (e.g., FIG. 3, item 31 "Unreachable Statement Column", col.3: 63 – col.4: 21; FIG. 4b, block 65, col.5: 29-67; FIG. 5, col.6: 52 – col.7: 8).

It would have been obvious to a person having ordinary skill in the art at the time the invention was made to combine Bates' teaching into Morshed's teaching. One would have been motivated to do so to detect an unreachable breakpoint and display a warning to a user as suggested by Bates (e.g., col.1: 61 – col.2: 37, emphasis added).

b) Morshed and Tsai (pp. 21-25):

As set forth in (7) above, Morshed explicitly teaches the claimed limitations.

In claim 21, Morshed does not explicitly disclose *monitors the computer program for the non-occurrence of any one of a second set of defined conditions, the non-occurrence of which violates the coding rule associated with said each of the pattern detectors.*

However, in an analogous art, Tsai further discloses *monitors the computer program for the non-occurrence of any one of a second set of defined conditions, the non-occurrence of which violates the coding rule associated with said each of the pattern detectors* (e.g., col.10: 58-67).

It would have been obvious to a person having ordinary skill in the art at the time the invention was made to combine Tsai' teaching into Morshed's teaching. One would have been motivated to do so to declare faults after a maximum wait threshold (maximum time to reach a specific breakpoint) and avoid the target program to hang indefinitely as suggested by Tsai (e.g., col.10: 58-67, emphasis added).

In conclusion, the examiner respectfully maintains the 35 USC 102 and 103 rejections over claims 1, 4-7, 11-12, 15, 17-19, and 21.

Claim Rejections – 35 USC § 102

9. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

10. Claims 1, 4-7, 15, and 17-19 are rejected under 35 U.S.C. 102(e) as being anticipated by Morshed (art of record, US Patent No. 6,721,941).

Claim 1:

Morshed discloses a program storage device readable by machine and *a software tool containing machine readable instructions stored on a physical medium for*

monitoring behavior of a running computer program (e.g., FIG. 12, a VM runtime system runs and monitors a program comprising Java byte code, col.21: 6-24; Monitoring DLL 414, col.21: 55-67)

for code patterns (e.g., FIG. 14, Java classes (code patterns) as best practice patterns, col.23: 1-64;

col.13: 40-56: "scope change" code blocks (other code patterns) as problematic coding patterns, which may create memory leaks, col.19: 13-33)

that violate a given set of coding rules (e.g., col.13: 43-49, coding rule as must free allocated memory to avoid memory leak;

col.17: 25-28: coding rules have been violated and run time errors occur;

col.20: 60-62: violating coding rules such as "improper parameter lists passed to functions, methods, or procedures, use of uninitialized variables";

FIG. 64, col.75: 63 – col.76: 23: results of violating coding rules displayed in a bug report)*, the software tool comprising:*

a pattern detector manager (e.g., FIG. 4, Code Instrumentation 50, col.8: 4-28; col.6: 63 – col.7: 2) *including machine readable instructions for inserting into a running computer program a plurality of entry breakpoints* (e.g., FIG. 14, blocks 442, 448, 452, 456, 460, 464, col.23: 1 – col.24: 11)*,*

automatically, with little or not intervention from a user (e.g., col.20: 14-19; col.20: 63 – col.21: 5)*,*

each of said entry breakpoints being associated with one of a plurality of defined coding patterns (e.g., FIG. 14, coding patterns of Java classes such as Method Entry (block 442), End of Method (block 446), Throw instruction (block 454), Method Call (block 462), Abort (block 448), col.23: 1-64)*; and*

a plurality of pattern detectors, each of the pattern detectors being associated with one of said defined coding patterns (e.g., FIG. 12, Instrumentation DLL 410, Monitoring DLL 414, col.21: 43-67, which instrument best practice patterns (Java classes) by associated instrumenting code as described in FIG. 14, col.23: 1-64*,*

or instrument problematic coding patterns ("scope change" code blocks) by associated instrumenting code to detect memory leaks as described in col.13: 40-56 and col.19: 13-33),

including machine readable instructions, and being invoked by the pattern detector manager, after one of the entry breakpoints associated with the coding pattern associated with said each of the pattern detectors, is reached in the computer program (e.g., FIG. 14, block 442 Instrument Method Entry, col.23: 6-13;

FIG. 15, details of the instrumentation for a method entry, col.24: 11-62)

for determining whether the computer program violates the coding pattern associated with said each of the pattern detectors (e.g., FIG. 14, block 448, Instrument for Abort to determine any violations, col.23: 14-22)

by inserting into the program at least one further breakpoint for identifying a respective step in the program that is part of the coding pattern associated with said one of the entry breakpoints (e.g., FIG. 14, block 448, Instrument for Abort as at least one further breakpoint associated with said method entry breakpoint in block 442, col.23: 14-22;

FIG. 16, details of the instrumentation for an abort step, col.24: 63 – col.25: 37;

col.66: 20-31, in the debugging mode "step", instrumenting entry breakpoint and subsequent breakpoints, col.66; 20-31;

FIG. 14, after Instrument Method Entry at block 442, inserting further breakpoints at blocks 452, 456, 460, 464, col.23: 36 – col.24: 11)*;*

wherein the plurality of defined coding patterns is selected from a group comprising best practice patterns (e.g., FIG. 14, Java classes (coding patterns) as defined best practice patterns may be instrumented and monitored, such best practice patterns including coding patterns such as method entry (col.23: 6-13), throw instruction (col.23: 44-51), exit point of said method (col.23: 52-64), ...) *and*

[selected from a group comprising] problematic coding patterns (e.g., col.13: 40-56, "scope change" code blocks (other coding patterns) as defined problematic patterns may be instrumented and monitored to detect memory leaks, and col.19: 13-33).

Claim 4:

The rejection of claim 1 is incorporated. Morshed also discloses *a debugger for debugging the computer program, and further including a launcher to invoke the pattern detector manager when the debugger is used to debug the program* (e.g., FIG. 12, col.21: 6-67).

Claim 5:

The rejection of claim 1 is incorporated. Morshed also discloses *the pattern detector manager removes the entry breakpoints at specified times* (e.g., col.20: 6-39).

Claim 6:

The rejection of claim 1 is incorporated. Morshed also discloses *the pattern detector manager removes the entry breakpoints and the further breakpoints at specified times* (e.g., col.20: 14-62).

Claim 7:

The rejection of claim 1 is incorporated. Morshed also discloses *the pattern detector manager includes means for monitoring for the occurrences of the entry breakpoints; and the pattern detector manager inserts said at least one further breakpoint into the computer program in response to the monitoring means detecting the occurrence of said one of the entry breakpoints* (e.g., FIG. 15-17, col.24: 11 – col.25: 67).

Claim 15:

Morshed discloses *a program storage device readable by machine, tangibly embodying a program of instructions executable by the machine to perform method steps for*

monitoring behavior of a running computer program (e.g., FIG. 12, col.21: 6-24, VM runtime system runs and monitors Java program; Monitoring DLL 414, col.21: 55-67)*, said method steps comprising:*

using a pattern detector manager to insert into a running computer program a plurality of entry breakpoints (e.g., FIG. 4, Code Instrumentation 50, col.8: 4-28; col.6: 63 – col.7: 2)*,*

each of said entry breakpoints being associated with one of a plurality of defined coding patterns (e.g., FIG. 14, col.23: 1 – col.24: 11; FIG. 15, col.24: 11-62; col.13: 40-56)*, and*

monitoring to detect the occurrences of the entry breakpoints in the computer program (e.g., FIG. 14, block 442, col.23: 1-64)*, and*

upon detection of one of the entry breakpoints in the computer program, further inserting into the program at least one further breakpoint for identifying a respective step in the program that is part of the coding pattern associated with said one of the entry breakpoints (e.g., col.24: 47-57; FIG. 17 and related text; col.66: 20-31)*; and*

using a plurality of pattern detectors for monitoring the computer program (e.g., FIG. 12, Instrumentation DLL 410, Monitoring DLL 414 and related text)*,*

wherein each of the pattern detectors are associated with one of said defined coding patterns, including the step of the program detector manager invoking each of the pattern detectors (e.g., FIG. 14, instrumenting a Java class with specific instrumenting code, col.23: 1-64; instrumenting "scope change" code block with specific instrumenting code, col.19: 13-33 and col.13: 40-56)*,*

after one of the entry breakpoints associated with the coding pattern associated with said each of the pattern detectors, is reached in the computer program, for determining whether the computer program violates the coding pattern associated with said each of the pattern detectors (e.g., FIG. 15, col.24: 11-62; col.1: 24-36; col.32: 61 – col.33: 27; col.55: 31-47)*.*

Claims 17-19:

Claims 17-19 recite the same limitations as those of claims 4-7, wherein all claimed limitations have been addressed and/or set forth above. Therefore, as the reference teaches all of the limitations of the above claims, it also teaches all of the limitations of claims 17-19.

Claim Rejections – 35 USC § 103

11. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

12. Claims 21 and 11-12 are rejected under 35 U.S.C. 103(a) as being unpatentable over Morshed in view of Bates (art of record, US Patent No. 6,839,893).

Claim 21:

Morshed discloses *a method of detecting code patterns in a computer program that violate a given set of coding rules, the method comprising the steps of:*

defining a set of coding rules (e.g., col.13: 43-49; col.17: 25-28; col.20: 60-62; col.75: 63 – col.76: 23),

each coding rule of the set of the coding rules being associated with a respective one pattern detector of a set of pattern detectors (e.g., FIG. 14, blocks 446, 450, 454, 458, 462, col.23: 1-64; col.13: 40-56; col.19: 13-33);

providing a pattern detector manager for managing said pattern detectors (e.g., FIG. 14, blocks 442, 448, 452, 456, 460, 464, col.23: 1 – col.24: 11; FIG. 12, block 410, col.21: 43-54);

providing a computer program, and running the computer program in a debug mode (e.g., FIG. 12,col.21: 6-67; col.23: 36 – col.24: 11);

the pattern detector manager identifying, during the running of the computer program in the debug mode, points in the computer program that relate to said coding rules (e.g., FIG. 15-17, col.24: 11 – col. 25: 67)*, and*

said pattern detector manager inserting into the computer program an entry breakpoint at each of said identified points (e.g., col.20: 40-49; col.20: 63 – col.21: 5)*;*

said pattern detector manager invoking each of the pattern detectors to monitor the computer program for a violation of the coding rule associated with said each of the pattern detectors (e.g., col.21: 6-67; col.13: 43-49; col.20: 60-62)*, including the step of:*

each of the pattern detectors inserting one or more further breakpoints into the computer program to identify further points in the computer program that relate to the coding rule associated with said each of the pattern detectors (e.g., col.24: 11 – col.25: 67; col.23: 14-22; col.66: 20-31)*, and*

tracking said additional breakpoints to determine whether the computer program violates the coding rule associated with said each of the pattern detectors (e.g., FIG. 14, col.23: 1 – col.24: 11; col.20: 6-62)*;*

wherein each of said additional breakpoints identifies a respective step in the computer program that is part of the coding pattern associated with said one of the entry breakpoints (e.g., FIG. 14, col.23: 1 – col.24: 11; col.24: 47-57; col.66: 20-31)*, and*

wherein each of the pattern detectors monitors the computer program for the occurrence of any one of the first set of defined conditions, the occurrence of which violates the coding role associated with said each of the pattern detectors (e.g., col.1: 24-36; col.32: 61 – col.33: 27; col.55: 31-47).

Morshed does not explicitly disclose *monitors the computer program for the non-occurrence of any one of a second set of defined conditions, the non-occurrence of which violates the coding rule associated with said each of the pattern detectors.*

However, in an analogous art, Bates further discloses *monitors the computer program for the non-occurrence of any one of a second set of defined conditions, the non-occurrence of which violates the coding rule associated with said each of the pattern detectors* (e.g., FIG. 3, item 31 "Unreachable Statement Column", col.3: 63 – col.4: 21; FIG. 4b, block 65, col.5: 29-67; FIG. 5, col.6: 52 – col.7: 8).

It would have been obvious to a person having ordinary skill in the art at the time the invention was made to combine Bates' teaching into Morshed's teaching. One would have been motivated to do so to detect an unreachable breakpoint and display a warning to a user as suggested by Bates (e.g., col.1: 61 – col.2: 37, emphasis added).

Claim 11:

The rejection of claim 21 is incorporated. Morshed also discloses *a debugger for debugging the computer program, and further including the step of invoking the pattern detector manager when the debugger is used to debug the program* (e.g., FIG. 12, col.21: 6-67).

Claim 12:

The rejection of claim 21 is incorporated. Morshed also discloses *the step of removing the entry breakpoints at specified times* (e.g., col.20: 6-39).

13. Claim 21 is rejected under 35 U.S.C. 103(a) as being unpatentable over Morshed in view of Tsai (art of record, US Patent No. 6,161,196).

Claim 21:

Morshed discloses *a method of detecting code patterns in a computer program that violate a given set of coding rules, the method comprising the steps of:*

defining a set of coding rules (e.g., col.13: 43-49; col.17: 25-28; col.20: 60-62; col.75: 63 – col.76: 23),

each coding rule of the set of the coding rules being associated with a respective one pattern detector of a set of pattern detectors (e.g., FIG. 14, blocks 446, 450, 454, 458, 462, col.23: 1-64; col.13: 40-56; col.19: 13-33);

providing a pattern detector manager for managing said pattern detectors (e.g., FIG. 14, blocks 442, 448, 452, 456, 460, 464, col.23: 1 – col.24: 11; FIG. 12, block 410, col.21: 43-54)*;*

providing a computer program, and running the computer program in a debug mode (e.g., FIG. 12,col.21: 6-67; col.23: 36 – col.24: 11)*;*

the pattern detector manager identifying, during the running of the computer program in the debug mode, points in the computer program that relate to said coding rules (e.g., FIG. 15-17, col.24: 11 – col. 25: 67)*, and*

said pattern detector manager inserting into the computer program an entry breakpoint at each of said identified points (e.g., col.20: 40-49; col.20: 63 – col.21: 5)*;*

said pattern detector manager invoking each of the pattern detectors to monitor the computer program for a violation of the coding rule associated with said each of the pattern detectors (e.g., col.21: 6-67; col.13: 43-49; col.20: 60-62)*, including the step of:*

each of the pattern detectors inserting one or more further breakpoints into the computer program to identify further points in the computer program that relate to the coding rule associated with said each of the pattern detectors (e.g., col.24: 11 – col.25: 67; col.23: 14-22; col.66: 20-31)*, and*

tracking said additional breakpoints to determine whether the computer program violates the coding rule associated with said each of the pattern detectors (e.g., FIG. 14, col.23: 1 – col.24: 11; col.20: 6-62)*;*

wherein each of said additional breakpoints identifies a respective step in the computer program that is part of the coding pattern associated with said one of the entry breakpoints (e.g., FIG. 14, col.23: 1 – col.24: 11; col.24: 47-57; col.66: 20-31)*, and*

wherein each of the pattern detectors monitors the computer program for the occurrence of any one of the first set of defined conditions, the occurrence of which violates the coding role associated with said each of the pattern detectors (e.g., col.1: 24-36; col.32: 61 – col.33: 27; col.55: 31-47).

Morshed does not explicitly disclose *monitors the computer program for the non-occurrence of any one of a second set of defined conditions, the non-occurrence of which violates the coding rule associated with said each of the pattern detectors.*

However, in an analogous art, Tsai further discloses *monitors the computer program for the non-occurrence of any one of a second set of defined conditions, the non-occurrence of which violates the coding rule associated with said each of the pattern detectors* (e.g., col.10: 58-67).

It would have been obvious to a person having ordinary skill in the art at the time the invention was made to combine Tsai' teaching into Morshed's teaching. One would have been motivated to do so to declare faults after a maximum wait threshold (maximum time to reach a specific breakpoint) and avoid the target program to hang indefinitely as suggested by Tsai (e.g., col.10: 58-67, emphasis added).

Conclusion

14. **THIS ACTION IS MADE FINAL**. See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the date of this final action.

15. Any inquiry concerning this communication should be directed to examiner Thuy Dao (Twee), whose telephone/fax numbers are (571) 272 8570 and (571) 273 8570, respectively. The examiner can normally be reached on every Tuesday, Thursday, and Friday from 6:00AM to 6:00PM.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Tuan Q. Dam, can be reached at (571) 272 3695.

The fax phone number for the organization where this application or proceeding is assigned is (571) 273 8300. Any inquiry of a general nature of relating to the status of this application or proceeding should be directed to the TC 2100 Group receptionist whose telephone number is (571) 272 2100.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

/Thuy Dao/
Examiner, Art Unit 2192

/Tuan Q. Dam/
Supervisory Patent Examiner, Art Unit 2192